An Easy Method To Accelerate An Iterative Algebraic Equation Solver

J. Yao

February 25, 2014

Journal of Computational Physics

An Easy Method To Accelerate An Iterative Algebraic Equation Solver

Jin Yao

Lawrence Livermore National Laboratory, California, USA

Abstract

This article proposes to add a single function calls to an iterative algebraic equation solver with an order n convergence rate, and to raise (for $n > 1$) the order of convergence to $2n - 1$.

INTRODUCTION

Newton's method for finding the root for $f(x) = 0$ is very widely used, both directly and as a conceptual basis for the development of further methods. There is a large literature on schemes to generalize the method to various higher orders. In particular, Kung and Traub[1] demonstrated that an equation solver with n functional calls can achieve an order of convergence 2^{n-1}. However it is believed that no equation solver that achieves this order has been constructed for $n > 4$.

Many of the existing fast equation solvers are skillfully constructed (for recent examples, see [4],[5],[6]) but we consider the simplicity of implementation. In this paper a simple idea is proposed which is to add a single extra function evaluation to an arbitrary one-point iterative equation solver of convergence order n, and thereby to accelerate the original scheme to an order of convergence $2n-1$. Furthermore, it can be shown that with each additional call of the derivatives, the order of convergence is raised by $n - 1$ more (see appendix).

THE ORDER OF CONVERGENCE

An iterative equation solver for a set of algebraic equations $\vec{F}(\vec{x}) = 0$ is said to have an order of convergence n when

$$|\vec{x}_{k+1} - \vec{x}_k| = O(|\vec{x}_k - \vec{x}_{k-1}|^n)$$

at the k^{th} iteration. An almost equivalent definition is that

$$|\vec{F}(\vec{x}_{k+1})| = O(|\vec{x}_{k+1} - \vec{x}_k|^n)$$

in the case when the Jacobian of the system is *non-zero* at the solution. The order of convergence of an iterative solver is a measurement of how fast it converges to the true solution.

The proposed new scheme accelerates an iterative solver with n^{th}-*order* convergence, with a single additional functional call, the order of convergence can be raised from n to $2n-1$.

There are two steps with this method and we demonstrate the procedure here for the case $n = 3$ with the well-known Halley's method[2]. Let x_k be the k^{th} estimate for the root. One solves the equation (assuming $f'' \neq 0$)

$$f(x_k) + f'(x_k)\delta + \frac{1}{2}f''(x_k)\delta^2 = 0, \tag{1}$$

and the two roots are explicitly expressed as

$$\delta = -\frac{1}{f''(x_k)}\left(f'(x_k) \pm \sqrt{f'(x_k)^2 - 2f(x_k)f''(x_k)}\right).$$

To recover Newton's method when the quadratic term vanishes, we pick only one root and it can be written as

$$\delta = \frac{\mathrm{sgn}(f'(x_k))}{f''(x_k)}\left(\sqrt{(f'(x_k))^2 - 2f(x_k)f''(x_k)} - |f'(x_k)|\right).$$

The above step uses three function calls. Note that a Taylor series for $f(x+\delta)$ at $x = x_k$ using Eq. (1) implies

$$f(x_k + \delta) = O(\delta^3). \tag{2}$$

The next step uses one more function call to gain two more orders of convergence. One adds a term $f(x_k + \delta)$ to Eq. (1), and solves

$$f(x_k + \delta) + f(x_k) + f'(x_k)\Delta + \frac{1}{2}f''(x_k)\Delta^2 = 0. \tag{3}$$

The solution is similar to that obtained in the first step:

$$\Delta = \frac{\mathrm{sgn}(f'(x_k))}{f''(x_k)}\left(\sqrt{(f'(x_k))^2 - 2(f(x_k+\delta) + f(x_k))f''(x_k)} - |f'(x_k)|\right).$$

Finally, let $x_{k+1} = x_k + \Delta$ for completion of the current iteration cycle.

One computes only four function values $f(x_k), f'(x_k), f''(x_k)$, and $f(x_k + \delta)$. However, the above scheme is fifth order convergent as shown next.

From a Taylor expansion one obtains

$$f(x_k + \Delta) = f(x_k) + f'(x_k)\Delta + \frac{1}{2}f''(x_k)\Delta^2 + \frac{1}{6}f'''(x_k)\Delta^3 + O(\Delta^4).$$

The sum of the first three terms in the right-hand side is equal to $-f(x_k + \delta)$ from Eq. (3); thus $f(x_k+\Delta) = -f(x_k+\delta)+f'''(x_k)\Delta^3/6+O(\Delta^4)$. However, from Eq. (1) and the Taylor expansion of $f(x_k + \delta)$, the above estimate becomes

$$f(x_k + \Delta) = \frac{1}{6}f'''(x_k)(\Delta^3 - \delta^3) + O(\Delta^4 - \delta^4) = (\Delta - \delta)O(\Delta^2, \delta^2). \qquad (4)$$

By subtracting Eq. (1) from Eq. (3) one arrives at

$$(\Delta - \delta)(f'(x_k) + O(\delta)) = -f(x_k + \delta) = O(\delta^3). \qquad (5)$$

It tells us that Δ and δ are of the same order and

$$(\Delta - \delta) = O(\delta^3).$$

One easily sees from Eq. (4) and Eq. (5) that

$$f(x_{k+1}) = f(x_k + \Delta) = O(\Delta^5).$$

Therefore the method is *fifth-order* order convergent; however it employs only *four* function values. The proof above can be generalized for arbitrary n.

If $f'(x) = 0$ at the solution, Newton's method either fails or converges slowly. The rapidly-converging scheme described above is more stable. However, the order of convergence is reduced from *five* to *four* because when $f' = 0$, Eq. (5) gives $(\delta - \Delta) = O(\delta^2)$ instead of $O(\delta^3)$. In practice if x_k is not close to the solution, the term under the square root $f'(x_k))^2 - 2f(x_k)f''(x_k)$ can become negative and break the iteration. In this case this term can be set to zero to keep the computation going.

If f' is finite at the solution, because $f \to 0$ when x_k is close to the solution, x^*, the term in the square root will be non-negative when sufficiently close to convergence; if $f' = 0$ at the solution, setting this term to zero (if it becomes negative) would give $\delta = -f'(x_k)/f''(x_k)$, which is similar to a Newton-method by L'Hospital's rule.

TO RAISE THE ORDER OF CONVERGENCE FROM n TO 2n – 1

An *one-point* iterative equation solver for $f(x) = 0$ can be generally written as a *fixed-point* iteration as

$$x_{k+1} = x_k + \delta(f(x_k), f'(x_k), f''(x_k)...) = x_k + \delta_k. \tag{6}$$

If the solver is of convergence order n, we also have

$$f(x_k + \delta_k) = O(\delta_k^n). \tag{7}$$

We are now going to show that the modified scheme

$$\bar{x}_{k+1} = x_k + \delta(f(x_k) + f(y_k), f'(x_k), f''(x_k)...) \tag{8}$$

has an order of convergence $2n - 1$ with $y_k = x_k + \delta_k$.

With a Taylor expansion of the above equation, one has

$$\bar{x}_{k+1} = x_k + \delta(f(x_k), f'(x_k), f''(x_k)...) + \frac{\partial \delta}{\partial f} \times f(y_k) + O(f(y_k)^2).$$

Because $f(y_k) = O(\delta_k^n)$, then to the order $2n$, one has

$$f(\bar{x}_{k+1}) = f(x_k + \delta_k + \frac{\partial \delta}{\partial f} \times f(y_k)) + O(\delta_k^{2n}). \tag{9}$$

Next we evaluate the partial derivative (evaluated at y_k) in the above equation. Because the original solver is n^{th}-*order* convergent has the truncated Taylor series

$$f(x) + f'(x)\delta(x) + \frac{f(x)''}{2}(\delta(x))^2 + ... + \frac{f^{(n-1)}}{(n-1)!}(\delta(x))^{n-1} = O((\delta(x))^n)). \tag{10}$$

Differentiate the above equation with respect to f, one finds that

$$1 + f'(x)\frac{\partial \delta}{\partial f} + f''(x)\delta\frac{\partial \delta}{\partial f} + ... + \frac{f^{(n-1)}}{(n-2)!}(\delta(x))^{n-2}\delta f = O((\delta(x))^{n-1}))\frac{\partial \delta}{\partial f}.$$

Collecting the coefficients of $\partial\delta/\partial f$, one realizes it is a truncated Taylor expansion of $f'(x + \delta(x))$ to the order $(\delta(x))^{n-1})$, so we can write

$$\frac{\partial \delta}{\partial f} = -\frac{1}{f'(x+\delta(x))} + O((\delta(x)^{n-1}). \tag{11}$$

Then one is able to estimate the size of $f(\bar{x}_{k+1})$ with Eq. (9) which is accurate to order $2n$.

$$f(\bar{x}_{k+1}) = f(y_k + \frac{\partial \delta}{\partial f} \times f(y_k)) + O(\delta_k^{2n}) = f(y_k) + f'(y_k)\frac{\partial \delta}{\partial f} \times f(y_k) + O(\delta_k^{2n}).$$

Because $f(y_k) = O(\delta_k^n)$ and $\partial\delta/\partial f$ is off from $-1/f'(y_k)$ by only $O(\delta_k^{n-1})$ with Eq. (11), the $f(y_k)$ term on the *right-hand-side* is canceled. One finally arrives at

$$f(\bar{x}_{k+1}) = O(\delta_k^{2n-1}). \tag{12}$$

Thus by adding a single function call $f(y_k)$, an arbitrary iterative equation solver Eq. (6) that is n^{th}-*order* convergent becomes $(2n-1)^{th}$-*order* convergent with Eq. (8).

NUMERICAL EXAMPLES

WITH A SINGLE NONLINEAR ALGEBRAIC EQUATION

An iterative equation solver of n^{th}-*order* convergence which solves the truncated Taylor expansion

$$\sum_{i=0}^{n-1} \frac{f^{(i)}(x_k)}{i!}(\delta_k)^i = 0, \tag{13}$$

with $\delta_k = x_{k+1} - x_k$, can be improved as before by re-solving the original equation but with an additional term:

$$f(x_k + \delta_k) + \sum_{i=0}^{n-1} \frac{f^{(i)}(x_k)}{i!}(\Delta_k)^i = 0 \tag{14}$$

and taking $x_{k+1} = x_k + \Delta_k$. This is worth doing when the evaluation of function values cost much more than solving polynomial equations (with an iterative solver, say).

Next we demonstrate the convergence rate of the new scheme for a *high-order* Taylor-expansion scheme

$$f(x_k) + f'(x_k)\delta + \frac{f''(x_k)}{2}\delta^2 + \frac{f'''(x_k)}{6}\delta^3 + \frac{f''''(x_k)}{24}\delta^4 = 0 \tag{15}$$

(which is *fifth-order* convergent) with an example $f(x) = \cos(x) - x$ which has a root at $x^* = 0.739085133215160641638918505.....$ Because a trigonometric function costs a lot to evaluate, it is desirable to use fewer function values for a specified accuracy. All numerical evaluations below are done with *mathematica* with an accuracy of 100 digits.

We take an initial guess that matches the root to *four* digits, $x_0 = 0.7388$; and the above *quartic* polynomial equation has the solution

$$\delta = 0.000285133215160641661631833398416 1...,$$

which gives $f(x_0 + \delta) = -1.05768 \times 10^{-20}$. This accuracy is expected because the order of convergence here is $5 = 20/4$, the ratio between digits of accuracy of two consecutive iterations, with this *fifth-order* Newton's scheme. Now we add the term $f(x_0 + \delta)$ to Eq. (15) and solve

$$f(x_0 + \delta) + f(x_0) + f'(x_0)\Delta + \frac{f''(x_0)}{2}\Delta^2 + \frac{f'''(x_0)}{6}\Delta^3 + \frac{f''''(x_0)}{24}\Delta^4 = 0$$

and expect the result, with $(2n-1)^{th}$*-order* convergence in theory, to have $9 \times 4 = 36$ digits of accuracy ($n = 5$ is the convergence order of Eq. (15), and 4 is the number of effective digits of x_0). Indeed, we find

$$\Delta = 0.000285133215160641655312087673873403313044341483 2...$$

and the modified solution $x_1 = x_0 + \Delta$ gives $f(x_1) = 1.17214333 \times 10^{-36}$. Therefore the order of convergence with the proposed scheme is numerically $36/4 = 9$, equal to $(2n - 1)$ just as proven in the previous section (with $n = 5$ here).

WITH A SYSTEM OF NONLINEAR EQUATIONS

For a system of equations, the most commonly used *root-finding* method is Newton's method; i.e., to obtain the solution of $\vec{F}(\vec{x}) = 0$, with $\vec{F} = (F_1, F_2, ...F_M)$ and $\vec{x} = (x_1, x_2, ...x_M)$ (M a positive integer) one solves

$$F(\vec{x}_k) + \vec{\nabla} F(\vec{x}_k) \cdot \vec{\delta}_k = 0 \tag{16}$$

and takes $\vec{x}_k + \vec{\delta}_k$ as the new estimate of the root. This scheme is *second-order* convergent. To accelerate the scheme, we add the term $\vec{F}(\vec{x}_k + \vec{\delta}_k)$ and solve

$$\vec{F}(\vec{x}_k + \vec{\delta}_k) + \vec{F}(x_k) + \vec{\nabla} F(x_k) \cdot \vec{\Delta}_k = 0 \tag{17}$$

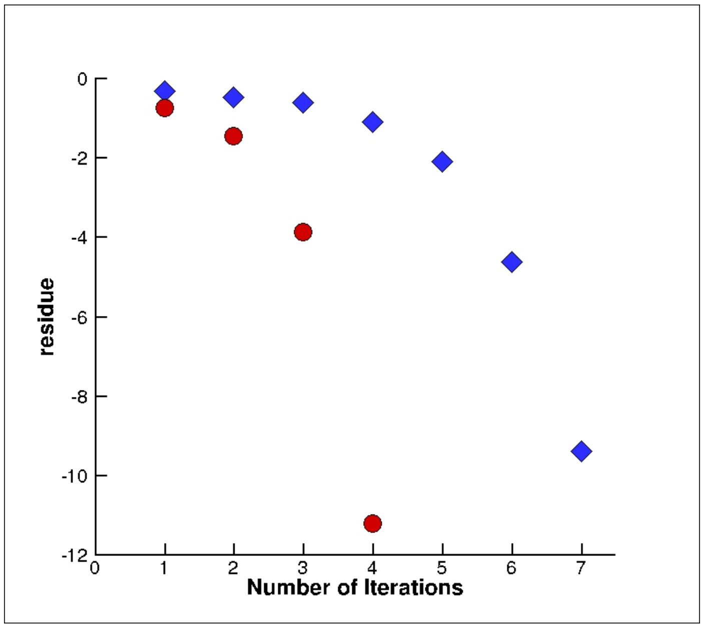

Figure 1: *The convergence map (residue logarithm vs. number of iterations) of Newton's method (diamonds, following a parabola) and the proposed method (circles, following a cubic curve) for the equation system Eq. (18).*

for Δ_k, and finally let $\vec{x}_{k+1} = \vec{x}_k + \vec{\Delta}_k$. The improved scheme is *third-order* convergent.

This convergence rate is obtained with no evaluation of *second* derivatives, but with M function evaluations in addition to the cost of the original Newton's method. Considering that the Newton's method requires M function values and M^2 derivatives, as M becomes large the improved scheme is considerably more efficient. The additional cost is relatively insignificant, but provides an extra order of convergence. Not only this, there is also no need to recompute the inverse of *Jacobian-matrix* in Eq. (17).

Finally we apply the new scheme to a system of algebraic equations

$$f(x,y) = \cos(\pi x)e^y + \sin(\frac{x^2+y^2}{2}), \quad g(x,y) = \sin(\pi y)e^x + \cos(\frac{x^2+y^2}{2}). \quad (18)$$

This system has a solution at $(x,y) = (1,0)$. With the initial guess $(x,y) = (0.7, 0.3)$, the solution can be obtained by taking either a Newton iteration scheme

$$f(x_k,y_k) + f_x\delta x + f_y\delta y = 0, \qquad g(x_k,y_k) + g_x\delta x + g_y\delta y = 0,$$

to solve for $x_{k+1} = x_k + \delta x$, $y_{k+1} = y_k + \delta y$; or with the scheme in this paper that

$$f(x_k+\delta x, y_k+\delta y) + f(x_k,y_k) + f_x\Delta x + f_y\Delta y = 0,$$

$$g(x_k+\delta x, y_k+\delta y) + g(x_k,y_k)) + g_x\Delta x + g_y\Delta y = 0,$$

to solve for Δx, Δy, and letting

$$x_{k+1} = x_k + \Delta x, \qquad y_{k+1} = y_k + \Delta y,$$

at the $(k+1)^{th}$ iteration.

The residues from both methods are plotted in Fig. 1. The proposed scheme, being *third-order* in this case, clearly converges much faster than the Newton method of *second-order* convergence.

CONCLUSION

For an iterative equation-solver that is n^{th}-*order* convergent, the order of convergence can be raised to $2n-1$ by adding a single *non-derivative term* to the expansion. The proof of this conclusion for an arbitrary iterative solver is given in this paper.

Many *high-order* iterative solvers [4],[5],[6] are special cases of the proposed scheme. When the proposed scheme is applied to Newton's method for a system of M equations, *third-order* convergence can be obtained with only M function evaluations, in addition to the $M + M^2$ function evaluations for the *second order* convergence obtained with the usual Newton's method. This indicates that a substantial speed-up on average over Newton's method can be achieved for $M \gg 1$ if the solution involves inverting the Jacobian.

APPENDIX: TO RAISE THE ORDER FROM $\mathbf{n}$ TO $\mathbf{(k+1)n-k}$

An iterative equation solver for $f(x) = 0$ can be generally written as a *fixed-point* iteration as

$$x_{k+1} = x_k + \delta(f(x_k), f'(x_k), f''(x_k)...) = x_k + \delta_k. \tag{19}$$

If a consecutive iteration is taken, one would obtain a better estimate of the root $\bar{x}_{k+1}$ with

$$\bar{x}_{k+1} = x_k + \delta_k + \epsilon_k, \tag{20}$$

where

$$\epsilon_k = \delta(x_k + \delta(x_k)).$$

If the solver is n^{th} order convergent we have $f(x_k + \delta_k) = O(\delta_k^n)$, thus

$$\epsilon_k = O(\delta_k^n), \tag{21}$$

this is easily seen with a Taylor expansion of $f(x_k + \delta_k + \epsilon_k)$ to a couple of terms. The refined estimate $\bar{x}$ would give a convergence order of n^2 such that

$$f(x_k + \delta_k + \epsilon_k) = O(\epsilon_k^n) = O(\delta_k^{n^2}) \tag{22}$$

Let's now consider the functions $g(x)$ and $h(x)$

$$g(x) = x + \delta(x) = x + \delta(f(x), f'(x), f''(x)..., f^{(n-1)}), \qquad h(x) = \delta(g(x)), \tag{23}$$

then analyze the behavior of

$$y(x) = g(x) + h(x). \tag{24}$$

From Eq. (22) one has

$$f(y(x)) = O(\delta(x)^{n^2}) = O(h(x)^n). \tag{25}$$

If we perturb $h(x)$ a little bit, from

$$h(x) = \delta(f(g(x)), f'(g(x)), f''(g(x)), ...)$$

to

$$\bar{h}(x) = \delta(f(g(x)), f'(g(x)) + d(f'(g)), f''(g(x)) + d(f''(g)), ...), \tag{26}$$

then ask how much $h(x)$ has changed, a Taylor expansion of δ at $g(x)$ gives that

$$\bar{h}(x) = h(x) + \frac{\partial h}{\partial f'} d(f'(g)) + \frac{\partial h}{\partial f''} d(f''(g)) + ...$$

We would like to evaluate the partial derivatives in the above expansion with Eq. (22). Taking the Taylor's expansion of Eq. (22) at an arbitrary $g(x)$ and using the definition of g, f, one sees that

$$f(g(x) + h(x)) = f + \bar{f}'h + \frac{\bar{f}''}{2}h^2 + ... + \frac{\bar{f}^{(n-1)}}{(n-1)!}h^{n-1} = O(h^n)), \tag{27}$$

where $\bar{f}' = df/dg, \bar{f}'' = d^2f/dg^2, ...$ Taking partial derivative to $\bar{f}', \bar{f}'', ...,$ with some algebraic manipulations one obtains

$$\frac{\partial h}{\partial \bar{f}'} = -\frac{h}{f'(y(x))} + O(h^n), \qquad \frac{\partial h}{\partial \bar{f}''} = -\frac{h^2}{f'(y(x))} + O(h^{n+1}), ... \tag{28}$$

In general one has

$$\frac{\partial h}{\partial \bar{f}^m} = \frac{h^m}{m! f'(y(x))} + O(h^{n+m-1}). \tag{29}$$

With Eq. (27), the above partial derivatives are obtained to the leading order. The expression of $\bar{h}(x)$ can then be written as

$$\bar{h}(x) = h(x) - (\frac{h}{f'(y)} + O(h^n)) \times d(f'(g)) - (\frac{h^2}{2f'(y)} + O(h^{n+1})) \times d(f''(g)) +$$

$$... - (\frac{h^{n-1}}{(n-1)!} f'(y) + O(h^{2(n-1)}) \times d(f^{(n-1)}(g)) + ... \tag{30}$$

We now consider how to approximate the derivatives of f about g in the expression of $h(x)$. Since we have all the derivatives of f about x obtained until the order $n-1$ already available, with these known derivatives we are able to do the following approximations:

$$f'(g) = f'(x+\delta(x)) = \sum_{i=0}^{n-2} \frac{f^{(i+1)}}{i!}\delta(x)^i + O(\delta(x)^{n-1}),$$

$$f''(g) = f''(x+\delta(x)) = \sum_{i=0}^{n-3} \frac{f^{(i+2)}}{i!}\delta(x)^i + O(\delta(x)^{n-2}),$$

until

$$f^{(n-1)}(g) = f^{(n-1)}(x+\delta(x)) = f^{(n-1)}(x) + O(\delta(x)).$$

With the above approximations we have $d(f'(g)) = O(\delta^{n-1})$, $d(f''(g)) = O(\delta^{n-2})$,... and so on. Substituting these in Eq. (30) one finds that

$$\bar{h}(x) - h(x) = -\frac{1}{f'(y)}\left(hO(\delta^{n-1}) + \frac{h^2}{2}O(\delta^{n-2}) + ...\right) + h.o.t.,$$

where $h.o.t.$ stands for *higher-order-terms*.

Because $h(x) = O(\delta(x)^n)$, the above estimate of the difference between $h(x)$ and $\bar{h}(x)$ can also be written in terms of δ only, such that

$$\bar{h}(x) - h(x) = -\frac{1}{f'}\left(O(\delta^{2n-1}) + O(\delta^{3n-2}) + ...\right) + h.o.t.. \tag{31}$$

This means, with our choices of approximations to the derivatives in $h(x)$, the modified estimate of the root

$$y_{k+1} = g(x_k) + \bar{h}(x_k), \tag{32}$$

would give

$$f(y_{k+1}) = O(\delta_k^{2n-1}). \tag{33}$$

The modified iterative scheme Eq. (32), with only one more function value $f(x_k+\delta_k)$ evaluated for $\bar{h}(x)$, would raise the order of an arbitrarily given n^{th}*-order* convergent iterative equation solver to $(2n-1)^{th}$ order.

However if we also evaluate $f'(x+\delta(x))$ exactly, then $d(f'(g)) = 0$, the first correction in Eq. (31), vanishes. Then with two more function calls $f(x_k+\delta_k)$ and $f'(x_k+\delta_k)$, the order of convergence of Eq. (6) would be raised to $3n-2$, and so on. The order raised by adding the first k derivative evaluations at $x+\delta(x)$ is $k(n-1)$ in general.

ACKNOWLEDGMENTS

The author would like to express appreciation to Dr. Robert Rieben, Dr. Brian Pudliner, and Dr. Frank Graziani of LLNL for reviewing this paper. The author also would like to thank Dr. David Miller of LLNL for detailed checking and insightful comments about this method. The author would like to express special appreciation to Dr. Lynda Lodestro of LLNL for proof-reading of the manuscript and expert technical comments.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

REFERENCE

[1]. H. T. Kung and J. .F. Traub "Optimal Order of One-Point and Multi-Point Iteration", *Journal of the Association for Computing Machinery*, Vol. 21, No. 4, pp.643-651, 1974.

[2]. Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes in C,* second edition, Cambridge University Press, (1997).

[3]. Weisstein, E. W. "Halley's Irrational Method", From *MathWorld* - A Wolfram Web Resource.

[4]. M. Grau and M. Nogurea, "A Variant of Cauchy's Method with Accelerated Fifth-Order Convergence", *Applied Mathematics Letters* 17, pp.509-517, (2004).

[5]. M. A. Noor, W. A. Khan, and A. Hussain, "A new modified Halley method without second derivatives for nonlinear equation", *Applied Mathematics and Computation* 189, pp.1268-1273, 2007.

[6]. C. Chun, "A simply constructed third-order modifications of Newton's method", *Journal of Computational and Applied Mathematics*, Vol. 215, Issue 1, pp.81-89, 2008.